



Monad – A New Command Shell

PDC⁰⁵
DEVELOPER POWERED

Jeffrey Snover, Architect
James Truher, Program Manager
TLN303
Microsoft Corporation

Top 10 Developer Calls To Action

Make an impression

☑ Follow the Windows Vista style guidelines	Look like a great Windows Vista app by using the common file dialogs, and providing high resolution live icons and preview controls for the shell
☑ Enrich the user experience	Use Windows Presentation Foundation to create engaging, scalable interfaces with multimedia, vector graphics and animation
☑ Enable users to visualize, search, and organize	Provide metadata for your file formats and enable search and organize in your applications

Be rock solid

☑ Run securely	Use least privilege user accounts and limit elevation requests; express privilege demands and limitations using Code Access Security
☑ Design for reliability and manageability	Use the new event logging and performance instrumentation capabilities for monitoring, troubleshooting, and analysis of your application; make it easy to deploy and update by using Windows Installer and ClickOnce
☑ Establish a customer feedback loop	Use the Windows Feedback Platform to log user errors, and Windows Developer Portal to analyze and fix them

Get connected

☑ Build for connected systems	Use Windows Communication Foundation to build connected systems that provide secure and reliable interoperability; take advantage of Windows Workflow Foundation to enable flexible workflow in applications
☑ Bring data to the user with RSS	Use the Windows Vista RSS common feedlist, shared data store, sync and parsing engine, and list extensions to RSS
☑ Make document data accessible	Use the same open XML file format as XPS and Office "12" to make documents easier for users to find, manage and integrate
☑ Build for mobility	Run great on laptops using power management and network awareness APIs; use advanced ink and gesture support for mobile user interfaces

Microsoft Management

Health Monitoring and Troubleshooting

- Windows Eventing 6.0
- Event Viewer 6.0
- Performance Counters 2.0
- Server Performance Advisor 2.0
- Microsoft Operations Manager

Administrative Control and Scripting

- Monad Shell (msh.exe)
- Task Scheduler 6.0
- Windows Management Infrastructure (WMI & WMIC)
- WS-Management and tools
- Command Prompt (cmd.exe)
- Microsoft Management Console 3.0

Deployment and Software Distribution

- Windows Server Update Services
- Installation Environment (Win PE)
- Windows Deployment Services (WDS)
- Systems Management Server

Configuration Management and Protection

- Group Policy Infrastructure
- Group Policy Management Console
- Windows Resource Protection (Files and Registry)
- User Access Protection

Roadmap

Scenario	Today	Tomorrow
GUI Shell	MMC	MMC with snap-ins layered on Monad
Command Shell	CMD	CMD SH CSH KSH Monad
Command Scripting	CMD scripting	CMD scripting Monad
COM Scripting	WSH <ul style="list-style-type: none">▫ VBScript▫ Jscript	WSH <ul style="list-style-type: none">▫ VBScript▫ Jscript Jscript (.NET) Monad

Monad Command Shell

● Motivation

- User experience
 - Preferred interaction model is Command-line oriented scripting
- Architectural
 - Refactor shell, commands, and utilities
 - Objectflow engine
 - Integrate command line, COM, and .NET scripting

● Inspirations

- Unix Shells and Utilities
 - Compositional model is extremely powerful
- AS400 / VMS
 - Naming and syntactic consistency aids in learning
- TCL / WSH
 - Embeddable & multi-language support
- BASH, PERL, C#
 - Style, utility, and expressiveness

ObjectFlow Engine

- Cmdlets emit objects
- Synthetic type system
 - Extendable by third-parties and users
 - Provides common data access syntax
- Engine manages pipeline objects
 - It shreds and coerces objects for parameter binding
 - It renders into textual views for interactive users or interoperability
- Utilities are object manipulators

Monad Value Proposition

- Developers write
 - .NET interfaces to management objects
 - Thin Cmdlets/Providers for a rich experience
- Monad exposes
 - Cmdlets => command line tools
 - Provider data => navigable drives
- Monad provides
 - Interactive shell
 - Strong language supporting
 - Command line, COM, and .NET scripting styles
 - Single parser/argument validator
 - Utilities and formatting

Cmdlet Development

- Focus on the experience of admin tasks
- Sub-class CMDLET
- Use attributes to expose properties as parameters
- Override one or more method
 - BeginProcessing(), ProcessRecord(), EndProcessing(), StopProcessing()
- Use helper functions
 - WriteProgress(), WriteDebug(), WriteObject(), WriteError(), WriteVerbose()
 - ShouldContinue(), ShouldProcess()
 - ThrowTerminatingError()



DEMO

Management Cmdlet

PDC⁰⁵
DEVELOPER POWERED

James Truher
Program Manager
AxP

Provider Development

- Focus on experience of interacting with your data
- Implement one or more abstract classes and interfaces
 - Abstract classes
 - ContainerCmdletProvider, DriveCmdletProvider, ItemCmdletProvider, NavigationCmdletProvider
 - Interfaces
 - IContentProvider, IPropertiesProvider, IPermissionProvider, etc
- Use attributes to expose capabilities
 - CmdletProvider, SupportsShouldProcess



DEMO

JET Database Provider

PDC⁰⁵
DEVELOPER POWERED

James Truher
Program Manager
AxP

A Command Scripting Language

- Scripting languages do not lend themselves to an interactive shell

```
$a = get-process          # vs Declaration
$a > filename            # Easy file interaction
copy-item a b            # No Quoting arguments
```

- Opportunity to improve learn-ability through better consistency

```
While ( ) { }            # vs while ... do ... done
If ( ) { }               # vs if ... then ... fi
Switch ( ) { }           # case ... ;; esac
```

- Provide production ready scripting

Tools For Discovery

- Get-Help
 - Retrieves documentation about Cmdlets and language topics
- Get-Command
 - Retrieves information about Cmdlets, scripts, aliases, functions, filters and native executables
- Get-Member
 - Retrieves information about *Objects*

Uniform Data Access Syntax

- XML

```
$x=[xml]"<a><b><c>TEST</c></b></a>"  
$x.a.b.c  
$x | format-custom
```

- WMI/ADSI

> \$x.Handles (msh) vs. x.Properties["Handles"] (vbs)

- ADO

- Column names mapped to properties

- Synthetic types created when possible

- System.__ComObject#{d30c1661-cdaf-11d0-8a3e-00c04fc9e26e}

- System.Xml.XmlElement#http://schemas.microsoft.org/root/HW_MNGT/2004/07#NumericSensor

- System.Management.ManagementObject#root\cimv2\Win32_Process

Scripting Text

- .NET String class is the foundation class
- Many .NET classes can be created with strings

```
[datetime]"1/1/2006"  
[xml]"<a><b>test</b></a>"  
[system.io.fileinfo]"C:\windows\notepad.exe"
```

- Native support for useful datatypes
 - Arrays, HashTable, Regex, XML
- Rich string operators
 - Match, Like, Replace, Multiply, Add

Scripting COM

- New-Object -ComObject <progid>
 - Binds to automation-compatible COM objects
- Reflect for properties/methods
- Access properties/methods
- Use in pipelines and utilities

Scripting .NET

- Create .NET objects
 - `$d=New-Object System.DateTime 2005,4,20`
- Inspecting/Accessing properties-methods
 - `$d.DayOfWeek`
 - `$d.AddMonths(6)`
 - `[DateTime]::Now`
 - `[DateTime]::IsLeapYear(2005)`
 - Method matching is done via least-type differencing
 - Use casts to override `[DateTime]::IsLeapYear([INT]$x)`

Abstract colorful lines in pink, blue, and yellow, flowing from the left side of the slide.

DEMO

All Together Now

PDC⁰⁵
DEVELOPER POWERED

James Truher
Program Manager
AxP

Call To Action!

- Write .NET interfaces to your management objects
- Expose tasks by creating classes which derive from Cmdlet
- Expose (hierarchical) data by implementing a Provider
- Write and share scripts

Community Resources

● At PDC

- For more information, go see
 - FUN311 Manageable Apps, FUN301 MMC 3.0, FUN315 Perf counters, FUN316 Schematized Events,
 - Hands-on-Labs: TLNHOL06 (cmdlet), TLNHOL08 (provider), TLNHOL25 (scripting)
 - Ask The Experts table
 - Tools and Languages Track lounge

● After PDC

- ScriptCenter:
<http://www.microsoft.com/technet/scriptcenter/default.mspx>
- Newsgroup: Microsoft.Public.Windows.Server.Scripting
- Channel 9 tag: <http://channel9.msdn.com/tags/Monad>
- Team blog: <http://blogs.msdn.com/monad/default.aspx>
- Wiki:
<http://channel9.msdn.com/wiki/default.aspx/Channel9.MSHWiki>
- 2 Monad “Best of TechEd” Webcasts
<http://www.microsoft.com/events/series/teched2005.mspx>

Microsoft[®]

Your potential. Our passion.[™]

© 2005 Microsoft Corporation. All rights reserved.

This presentation is for informational purposes only. Microsoft makes no warranties, express or implied, in this summary.